Attorney Docket No. 9637-000006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

PATENT

Inventors:	Paul Lawlor	)
Serial No.	09/705,361	)
Filed:	November 3, 2000	)
For:	**PACKAGING PERISHABLE PRODUCTS**	)

TRANSMITTAL OF PRIORITY DOCUMENTS

Hon. Commissioner of Patents & Trademarks
Washington, D. C. 20231

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on December 20, 2000.

By Christopher M. Brock

Sir:

Pursuant to the provisions of 35 U.S.C. 119, enclosed herewith is a certified copy of Great Britain Patent Application No. 99 26 198.4, filed November 5, 1999, as identified in the Declaration of this application. In support of Applicant's priority claim, please enter this document into the file.

Respectfully submitted,

Dated: December 20, 2000

By: Christopher M. Brock
Christopher M. Brock
Reg. No. 27,313
Gregory A. Stobbs
Reg. No. 28,764
Attorneys For Applicant

HAYNES BEFFEL & WOLFELD LLP

GAS:pal

The Patent Office

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

I also certify that the attached copy of the request for grant of a Patent (Form 1/77) bears an amendment, effected by this office, following a request by the applicant and agreed to by the Comptroller-General.

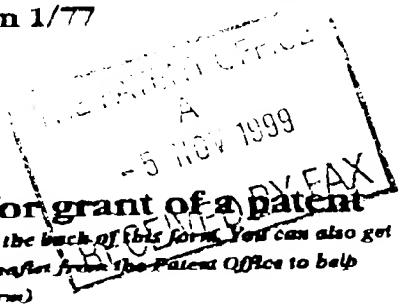
In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Patents Form 1/77

Patents Act 1977
(Rule 16)



The Patent Office

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office
Cardiff Road
Newport
Gwent NP9 1RH

1. Your reference KJB/sjp/CLARICOM 1 — 05 NOV 1999 — **9926198.4**

2. Patent application number *(The Patent Office will fill in this part)*

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

CLARICOM LTD
10 HEATHCOAT BUILDING
NOTTINGHAM SCIENCE PARK
UNIVERSITY BOULEVARD
NOTTINGHAM NG7 2QJ

Patents ADP number *(if you know it)* 7775024001

If the applicant is a corporate body, give the country/state of its incorporation

4. Title of the invention

IMPROVEMENTS IN PRINTER SYSTEMS

5. Name of your agent *(if you have one)*

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

WITHERS & ROGERS
GOLDINGS HOUSE
2 HAYS LANE
LONDON
SE1 2HW

Patents ADP number *(if you know it)*

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of

YES

Patents Form 1/77

Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form —

Description 34

Claim(s) —

Abstract — 3

Drawing(s) 2

10. If you are also filing any of the following, state how many against each item.

Priority documents —

Translations of priority documents —

Statement of inventorship and right to grant of a patent (Patents Form 7/77) —

Request for preliminary examination and search (Patents Form 9/77) —

Request for substantive examination (Patents Form 10/77) —

Any other documents (please specify) —

11. I/We request the grant of a patent on the basis of this application.

Signature Date

12. Name and daytime telephone number of person to contact in the United Kingdom KARL SUKENFATHER.

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

(3)

Improvements in Printer Systems

The invention relates to printers and printer systems, in particular coders or other printers used on production lines, such as for fast moving consumer goods such as food stuffs, for printing on packaging.

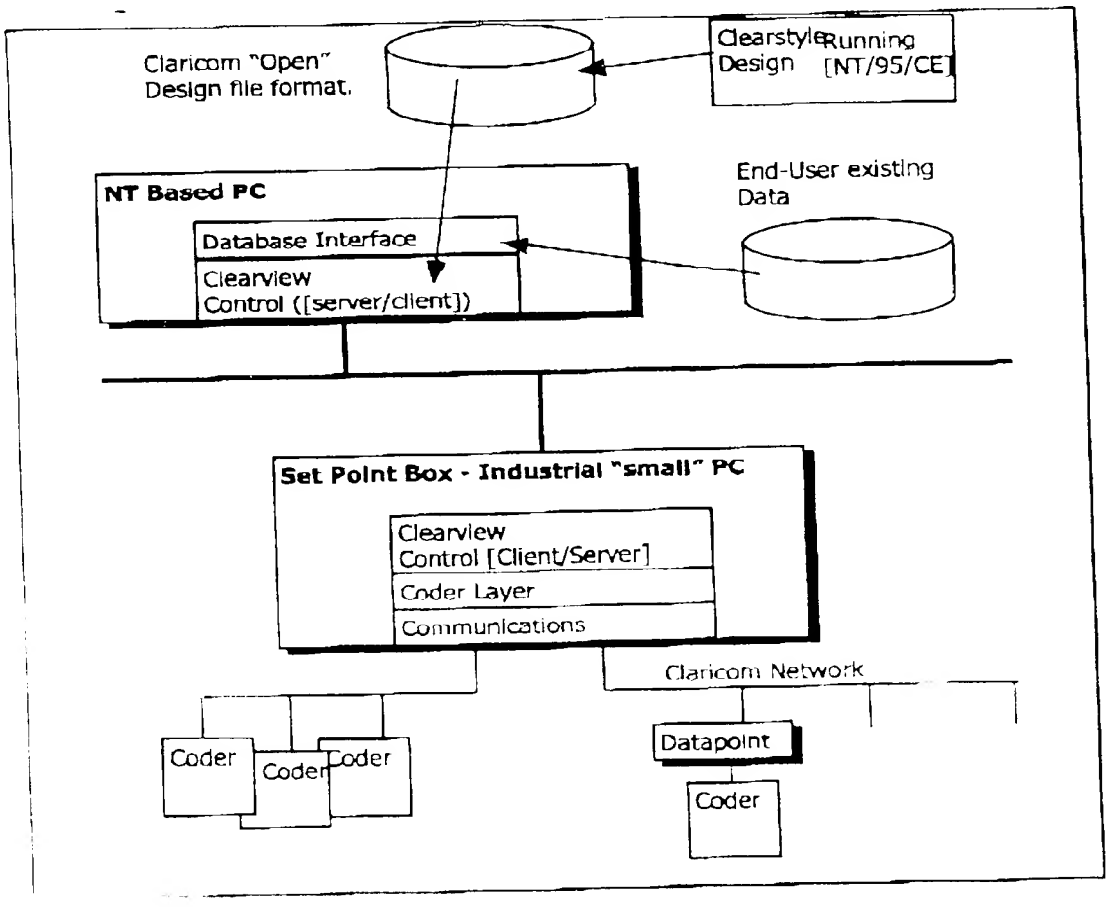
Objects of the invention include:

- centralised control of multiple coding and labelling devices in a common way;
- a solution across the wide variety of vendors and coding technologies used, independent of coder manufacturer;
- a bridge for the current communications gap between a company's I.T. infrastructure and the coding equipment on the factory floor.

Providing this solution reduces the considerable waste and product recall costs suffered by manufactures due to errors in the printed sell-by-dates, batch codes and bar codes.

Invention Overview

At the broadest level, and from an end-user marketing perspective the System of the invention products can be shown as follows:

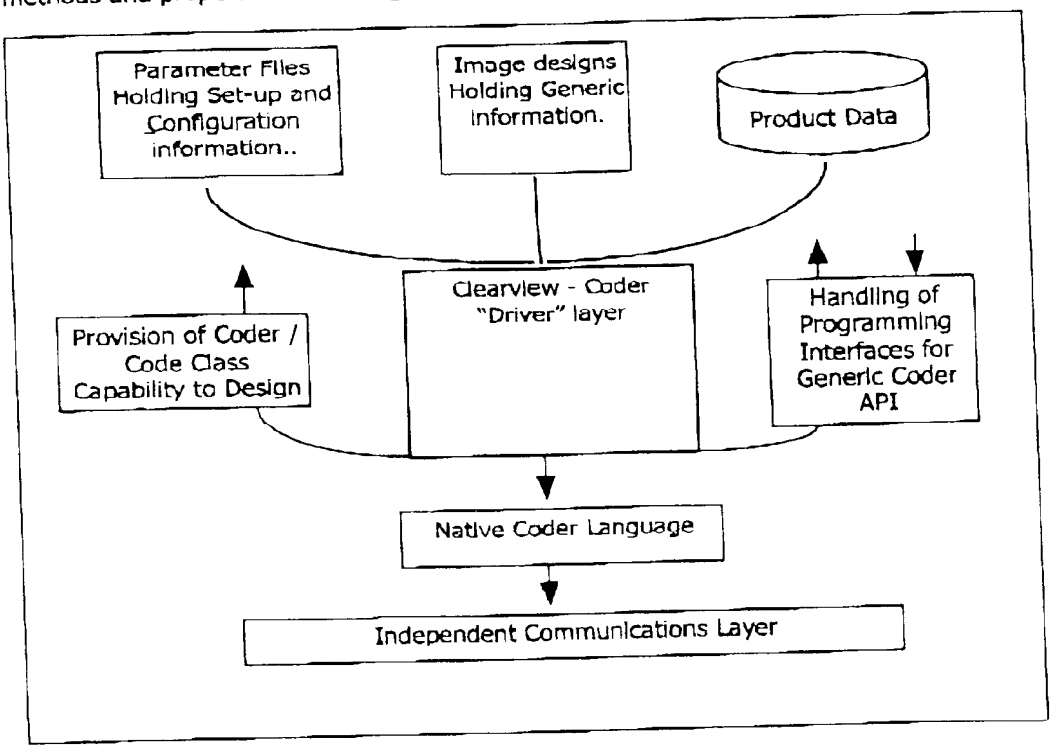


The core of the solution is the day-to-day coder control product **Clearview**[1] this product will either run on a desktop PC or in an Industrial PC (**Setpoint**). Coder networking is achieved using the coder's existing networking capability, where possible, or using a The Invention supplied networking solution; the solution will include RF, Ethernet or RS485 as the transport medium. To allow the Invention network to be used with any coder conversion boxes are used to provide the necessary electrical and protocol conversion (**Datapoint**).

An image design and data management application is also needed (**Clearstyle**). This provides device independent information to allow the "best-fit" information to be sent to a given Coder by Clearview.

The other strategic point illustrated in the figure is the ability to provide Clearview in both Client and Server modes. This separation allows centralised monitoring and control of multiple Setpoint networks from any PC on the Network.

The following diagram shows the overall Invention solution from a data perspective. This shows the need for a generic means of storing information and the need for a generic set of methods and properties for dealing with the wide variety of coders available.



The Clearview Coder "Driver" layer is shown as a single component, however it is possible that elements of the logical driver functionality are split into different components. Doing this allows optimisation of the amount of software required to run in a Setpoint box or ultimately in a coder itself.

The generic image design information needs to store information about how an image was created, for example, was the image optimised for a specific coder or class of coders, or was

[1] All product names Claricom, Clearview, Setpoint, Clearstyle and Datapoint are trade marks of Claricom Ltd and represent the system or elements of the system according to the invention.

the design purely generic. Sections of the design, or parameter files may hold information about design choices made by the user to determine the required method of overcoming limitations of the particular coder.

1.1 Further Objectives

A number of architectural technical objectives have been drawn-up. The goals are listed below, and will be echoed by specific areas in the remainder of the specification:

1. Provide a single application for controlling a wide variety of in-plant coder technologies from any vendor. *A key point - this is **the** end-user customer requirement - an end-user will focus on this point alone. For Coder Companies the elegance/attractiveness of the technical solution to provide future capabilities is very significant.*

2. The application shall have a number of independent layers, e.g. A control layer (the programmable graphical representation of the plant with control button(s) etc.), A coder specific layer (API) and a communications layer. Any customer may choose to only use some of the layers, for example, a SCADA application will only use the coder specific and communications layers. *Providing other industrial control solution providers with a common/broad coder control offering will increase the market coverage possible by Claricom. The architecture must allow a scaleable product and therefore price offering to maximise returns from all markets.*

3. Provide a common programming interface for a wide variety of in-plant real-time coders. *The Coder Application Programmers Interface (API) layer is the key to providing a common approach for a wide range of printers. It may be that the API only defines certain functions for certain classes of printer technology, e.g. do not expose PDF417 (complex 2D bar codes) for continuous inkjet coders. The common interface will be used by the Invention Control Application layer and will be key to providing a valuable solution to other industrial solution providers such as SCADA companies. For example, an OPC compatible interface can be provided for SCADA and MES applications to access product counts and coder status.*

4. Provide an open file format for defining the style of what is to be printed and where the data is to come from. *This approach will limit end-user reluctance to adopt yet another proprietary method of defining what needs to be printed.*

5. Wherever technically possible use existing standards for the file format for defining the style of what is to be printed and where the data is to come from. The file format will provide the ability to add programmed data handling e.g. provide a software module containing specialist batch code calculation and rounding. *Initial investigations indicate that XML/XSL, or building on top of HTML, can be used as a framework. (See section 1.12, "XML as the Chosen Standard").*

6. The Control layer will provide its own data handling functions and provide the option of linking to other sources of data using industry standard methodologies (SQL, ODBC, COM, OPC where applicable). *This top layer of the application must provide a wide range of "canned" options, similar to the built in database that many label design applications provide. A built-in solution to allow Works Order style working ("Next") will be provided.*

7. The data handling functions will allow the incorporation of applications to provide highly specialist functionality. *To allow linking into systems with highly specialist requirements the ability to specify custom software modules to process information prior to being downloaded to a printer will be provided.*

8. Provide Client/Server functionality, e.g. A PC has one copy of the Control Application to show the status of the Control Unit for each line (other PCs, possible Windows CE). These

control units are, in turn, connected to the coders. *In some applications a small industrial panel PC will be required for line configuration and to communicate with the coders for that line. The Control application must provide [Client/Server] (line PC) and [Server/Client] versions (Production Managers Office PC). The [Server/Client] version will allow the status and performance of each line to be monitored and allow the user to drill down to view the status of individual coders on that line. If the performance is acceptable this may be possible by viewing a dynamically updated Web page that represents the status screen of any Client.*

9. The Client/Server User Interface shall be programmable using HTML as the graphical design and command button framework. Canned controls will be provided to allow the most common requirements to be produced without specialist programming knowledge. *Users and Control Solution providers must be able to tailor the user interface that the Invention Control Application presents. As far as possible a Web Page approach will be used, as long as performance is not compromised. The canned controls will be provided through the use of Active Controls (COM modules, or Java Applets).*

10. It shall be possible for a third party to generate their own specialist routines for incorporation into the Control interface, e.g. plant specific interfacing to PLCs etc. *By using a Web Page approach users will be able to choose the style of programming used depending upon the performance required and the skill level of the available programmers. Plus, if speed of operation is paramount the user always has the option of using the API interface.*

11. The communications layer will allow for both OPEN communications methods AND the use of vendor specific legacy networking methods. *Handling legacy applications will be an issue for Invention. Customers may be resistant to scrap a working network. Providing a communications layer that uses a common interface back to the printer layer will allow vendor networks, where the protocol information is available, to be accommodated.*
12. Where ever possible existing technology will be used to provide the functionality required. *E.g. use Excel or Access macros to generate customer reports rather than producing an in-house report generator.*

System elements

1.2 Clearstyle

1.2.1 Definitions

Clearstyle is the working title for the image design element of the sytem according to the invention. Whilst Clearview will be the central focus of the Clarlcom solution the design element Clearstyle provides the focus in terms on the overall concept of the final solution.

The preferred elements are that Clearstyle allows the end-user to produce generic designs that can be used with any coding device (ideally) or at the very least with a particular class of coders e.g. small character Inkjet.

Clearstyle will use a published and commonplace standard document format as the basis for storing the image information and indicating the source of data for particular items on the image. It is appreciated that extensions to any current known standard will be needed to incorporate the needs of real-time in-plant coding equipment. The requirements of this file format are discussed further in section 1.3.

It is envisaged that Clearstyle will be used away from the factory floor; however the design of this product should allow a version to run with the hardware/operating system platform used for the Setpoint box (see section 1.6).

1.2.2 Objectives

To provide:

- device independent image design and data management;
- functionality that is aimed towards the specific needs of the package coding market;
- the ability to add new, or customer specific functions through the use of active controls; such controls to then be used to provide the "real" functionality at print-time.
- an "out-of-box" database solution for users where information is not already available on the network.

1.3 Coder Independent File Format (CIFF)

1.3.1 Definitions

To meet the objectives of Clearstyle a file format for the image design information is required; this Coder Independent File Format will be based on a current OPEN and EXTENSIBLE file format; XML is clearly a contender for this job.

In time this format will be published, thus allowing any design package to be provided with a CIFF driver and allow system integrators to produce and manipulate these design files.

Given the requirements of the real-time in-plant coding market it is deemed preferred that the file format provides methods for adding specialist functionality using Active components. It should be possible to create these components in a variety of languages to suit the needs of the application.

1.3.2 Objectives

The key functions of the CIFF will be to define:

- the properties of the image (size, class of printer the image was created for etc);
- links to any relevant coder configuration information, such as print performance parameters etc.;
- the number of objects placed on any given image;
- the properties of each object, e.g. position, size, orientation, font, barcode type, etc.;
- the source of the data to be used to print the required image. This could be user input, data from an ODBC data source or information from an active component.

1.4 Clearview

1.4.1 Definitions

The central focus of the Claricom product offering will be the Coder Control and Networking product that will be used by line operators on a day-to-day basis.

It is preferred that Clearview is highly modular and takes advantage of the latest distributed

From an end-user perspective it is this product that is used TO control what is printed and provides the "junction" between the various coders used and factory control and other I.T. infrastructure and must provide the flexibility needed to fit-in with the end-users' methods of working.

Clearstyle needs to run in two places: on an industrial PC (Setpoint) on the factory floor; or on a desktop PC located away from the lines where the coders are located; or as a combination of the two (Client/Server operation).

1.4.2 Objectives

To provide:

- an "out-of-box" solution for networking in-plant, real-time coding equipment, including industrial ink-jet (CIJ, DOD, Piezo), Thermal Coders and Label printers, Laser coders;
- a common interface for multiple coders from a wide variety or coder manufactures and coding technologies;
- a user with a single point of data entry to allow a production line, including differing types of coding equipment to be set-up with a single operation;
- a programmers interface (Coder API or CAPI) using state-of-the art distributed component methods (COM/DCOM) to allow other applications, such as SCADA and MES systems to automatically set-up the coding equipment on a line. This programmable interface must be generic across the full range of coders supported;
- a highly programmable user interface - using HTML or similar technology to allow active components to be added to any given customer solution;
- a number of independent layers to make incorporation of new coders, or new methods of communications as simple as possible.

1.5 Coder Application Programmers Interface (CAPI)

1.5.1 Definitions

The CAPI is really a part of Clearview and simply exposes the functionality of Clearview for use by third party developers. The interface will be provided using the latest component based techniques, e.g. COM/DCOM.

The interface will provide control, set-up, data management and maintenance functions. Status information will be provided using exception based reporting (e.g. no requirement on the calling application to poll for status).

No format creation functionality will be provided by this interface; the open file format (CIFF) will provide this.

1.6 Setpoint

1.6.1 Definitions

The Setpoint box will be used to provide the factory floor interface for setting up a given line. This unit will run Clearview [Client/Server] and provide a local point of connection for the coders in a given area. The Setpoint box with then provide Ethernet connectivity back to the wider I.T. network installed within the end-user site; this may include connecting back to a "[Server/Client]" version of Clearview.

1.6.2 Objectives

To provide:

- a robust, cost effective point for connecting multiple coders;
  Note THAT in-order for the Setpoint box to be "cost-effective" it is preferred that the software architecture reflects this need, e.g. RAM requirement;
- the ability to support a number of differing transport mechanisms for coders (RF, RS485, RS232, Ethernet etc);
- the ability to link back to the wider office/factory network using standard TCP/IP Ethernet networking;
- a graphical user interface using touch screen or other robust and simple means of user interaction.

1.7 Datapoint

1.7.1 Definitions

Many coders that are in use in today's factories will not provide any form of built-in networking support. Datapoint allows Claricom to provide a networked solution for such coders. To provide this a number of protocol converters are needed to provide the appropriate electrical and software protocol conversion.

1.7.2 Objectives

To provide:

- a cheap and reliable solution for networking legacy coding equipment that only provides simple RS232 communications or proprietary RS485;
- a number of options to support a variety of transport mechanisms.
- Clearstyle

Clearstyle is the image design and data management section of the Claricom offering. Users of Clearview will use Clearstyle to define:

- what the printed images need to look like for each level of packaging;
- where the data for each item comes from, or how it is calculated;
- the framework for adding additional programming to a given item on an image.

To do this Clearstyle must deal with the following types of information in a generic, coder independent fashion:

- image design information, number, type, position and style of information on an image;
- product related data, either as part of the Clearstyle product or via links to existing customer information.
- set-up information (or coder parameter information) for the printing process in relation to a particular product, e.g. the print darkness etc.

Clearstyle must be able to deal with the wide range of coding devices used from primary package coding through to pallet labelling. To accommodate the wide differences in these coders Clearstyle will include the concept of a Class of printers, classes would be Small Character Inkjet, Large Character and so on. This concept would give a user the option to design for:

- a particular coder, e.g. "I only have one type of coder in-plant why do anything else?"
- a class of coders, e.g. "I have a mix of Domino and Videojet small character inkjet coders and want to create a single design that will work for both";
- a generic coder, e.g. "I have some inkjet and some thermal overprinters and want the same design to work on them all".

Given the differences in capability and performance of these devices today compromises will need to be made when converting generic coder, or coder class, image designs into the native language of a specific coder. To provide the user with the best possible information about what will be printed Clearstyle will provide the following functions:

- generic WYSIWYG, e.g. the ideal output if the coder can support all the features, fonts and bar codes used on a the image design.
- coder class WYSIWYG, Clearstyle displays what a coder of the selected class would print, this would use the lowest common denominator of features.
- coder specific WYSIWYG, Clearstyle displays what the selected coder would print.

To provide the user with the best possible information when creating a generic design Clearstyle will provide an option to provide detailed information about the compromises made when converting an image design for a particular coder or class of coders. For example, a PDF417 bar code is selected for a coder that does not provide internal support for PDF417, if the resolution of the coder permits, the barcode could be provided to the coders as a graphic as long as the code is within the size constraints of the coders print head, the choice made and the options available must be provided back to the user. In addition concessions will need to be made to the physical head arrangement on certain types of coders, for example, the allocation of sections of an image to specific print heads on multi-head coders.

1.8 Operator Functionality

In general the user interface will:

- use common practice of Microsoft Windows NT products;
- where common practice makes a particular function difficult for a novice Window's user an alternative more obvious method should be provided, preferably in addition to the common practice method;
- provide functions commonly required within the product coding area as easily accessible functions, e.g. no "macro-programming" should be required to perform features such as:
  - offset dates, for use as best-before-end and sell-by dates;
  - merging of data from multiple objects on an image into a single object, such as a bar-code;
  - extracting sections of data from objects for inclusion in other fields, this facility to be combined with the merging function;
- provide advanced features that allow the base "command string" based data language to be accessed, e.g. manually enter a merge string vs. using the menu system to build the string;
- use as few layers of menus as practically possible;
- provide separate, easily edited, resources to allow the interface to be translated into other languages, or for use in OEM versions;
- provide a multiple document interface if this does not make the interface over cluttered;
- limit parameter values within practical ranges, e.g. do not allow combinations or parameters that are not allowed;
- wherever possible provide "Wizards" to obtain all the required information for a particular task, and deal with related parameters;

- provide default values that are logical to the end-users e.g. the default size for an EAN-13 bar code is 100%.

Clearstyle will provide a WYSIWYG image design environment.

- all functions should be possible from the WYSIWYG window. It shall be possible to:
  - add objects to the design;
  - delete objects from a design;
  - change the position;
  - change the orientation[2];
  - Ideally, provide on-screen editing of data;
  - allow the properties of an object to be edited, e.g. font, bar-code type source of data.
- provide print-preview options to allow a particular generic, or coder class, design to be viewed as it will be printed on a particular coder, including previewing a particular product;
- hints will be provided to inform a user where a design choice will mean that the performance of any specified class of coders, or specific coder will be compromised (e.g. the core ability to print the image, the update time and therefore the maximum throughput and the download time for the image);
- it shall be possible for the source of the data from any object used on an image design to be specified (see section 1.8.2.).

In addition, Clearstyle will provide a data management capability. This shall be easily accessible from the main design window and as an independent application further details are provided in section 1.8.2.

1.8.1 Coder Independent Design of Image

The WYSIWYG design environment must be possible to provide for:

- totally generic design, e.g. provide no feed-back based on specific coder capability;
- design for a class of coders, e.g. work within the lowest common denominator for the specified group of coders, such as small character CIJ;
- design for a specific coder, here the area and capabilities would be restricted to those of the selected coder.

Regardless of the method used for arriving at the final design the information will be stored in a generic fashion to allow the best-possible job to be made of printing the design on another coder.

To allow Clearstyle to handle provide WYSIWYG representation for images for specific coders, or classes of codes the printer drivers must provide the following information:

- maximum supported image size;
- text positioning capability (absolute/relative[3]);
- fonts supported, including bit-map and scaleable fonts;
- does the coder support downloadable bit-map and/or scaleable fonts?

---

[2] Depending upon the type of target coder selected, coder supported orientations may only provide internal support for 0°,90°,180° and 270° or only 0°.

[3] Note: certain types of coders, particularly small character inkjet coders do not support [illegible] positioning of objects. Support is provided for "n" lines of text with control

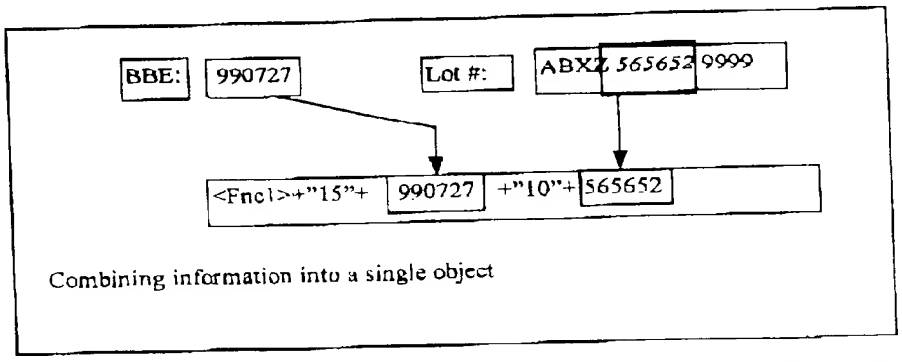
- graphics capability, maximum size, number etc.
- bar-code support;
- real-time features, e.g. what features are provided by a native coder command;
- multi-head support, e.g. MARKEM 5000 or print and apply with differing label front and side;
- hidden fields;

1.8.2 Data Management

1.8.2.1 Data Sources

Clearstyle will allow the user to specify where the data for a specific object on an image design comes from. The following sources will be provided:

- data is set by the image design, e.g. company name, fixed information such as "LOT:";
- Clearstyle database objects, here the data will come from Clearstyle's automatically generated database (see 1.8.2.2 and 1.10);
- operator entered information where data is entered when the image is selected for printing; this capability must operate over a group of images for different types of coders (see notes below);
- automatically generated objects such as date, time, incremental objects etc. Where a coder does not support a "generic" feature internally Clearview will provide the intelligence;
- machine resident information[4], such as machine number, line code;
- information that is combined from other objects and manually entered information, including the ability to merge a sub-set of an objects, e.g.



Combining information into a single object

This capability must use be provided without recourse to macro style programming by the user.

- all objects must be able to obtain base information from external data sources; this includes graphical information as filename or Binary object, the data sources must include:
  - ODBC data sources, including SQL, using latest Microsoft supported techniques such as ODBC drivers and, where appropriate, ADO (DAO);
  - data from active components, to include as a minimum COM/Active-X.

[4] It is appreciated that provided access to such information will be difficult as the methods used by individual coders will vary wildly. It is suggested that the coder driver layer provides information back to the design layer about what information is available for a coder/coder class etc.

- note that Clearview must provide methods for selecting the required row of data from record based data structures and provide options for testing that such data links are operating correctly during print-preview;
- pre-programmed active components[5] shall be provided to give packaging specific functionality such as SSCC calculations specialist check-digit calculations and rounding of dates etc.

1.8.2.2 Automatic database functions

In some cases an end-user will not have existing databases containing the information required for printing, or the information that is available is not in the correct format or not permanently available.

To provide a solution, in these cases, the Claricom suite of software has to provide a built-in database capability.

The users of this database will typically, not be IT specialists and it the preferred therefore that as many database creation and maintenance functions as possible are automated.

The database capability must provide:

- automatically generated database that uses the specified image design to create the required database fields;
- OR, select an existing database;
- when adding a automatic database object to a design allow an existing database field to be selected for the object OR allow a new field to be added;
- flexibility in the definition of a key field such that an object on the image design can be selected as the key field;
- allows data to be manually imported from other database applications;
- provides support for cut, copy and paste;
- provides a master index for product selection regardless of the base image design or coder selected.

The database structure must be flexible enough to support the following styles of application:

1. Line set-up where a single product selection must provide the data required for each stage of the packaging process, by differing coders;
2. "Next" style product selection where the list of product coders set to run on a given day is determined by a supervisor and key traceability information fixed. On a product change over the operator choices from the list of pre-selected product and may entered other traceability information only available at run-time;
3. "Tree" style product selection where the user drills down to the required product by using logical groupings.

In all three cases the Claricom database structure may need to link into other sources of data. The ability to provide additional keys for linking to other data tables via ODBC etc. is therefore preferred.

The complexities of the product selection methods will not be significant to Clearstyle and the database creation functions. It is Clearview that will need to implement the different product selection methods these can be provided as distinct components that use the core data

[5] Note: these components should be written such that they can be used to provide both design-time and run-time capability, e.g. the same component can be used by Clearstyle and [illegible] product selection

structures generated by Clearstyle. By splitting this functionality off into a component the ability to provide customised product selection becomes an option.

1.8.3 Definition of Connections to other sources to data.

Clearstyle shall allow the data for a field to be provided by other databases. Standard methodologies such ODBC, ADO, SQL must be supported.

Character set mapping will be provided, e.g. ANSI to Unicode etc. to allow data to be used from source without forcing the end-user to use alternate character codes.

1.8.4 Definition of Connection to other "Active" components

It shall be possible for a object on an image to obtain its data, or the actual rendered image from an active component. Any additional parameters required by the component shall be obtained using a common Component interface.

Typical applications for such components would be:

- a specialist control to obtain data from a check-weigher or PLC.
- to add a new bar code symbology;

The "active component of the coder is described in more detail in Section 11 and shown in figures 1 and 2.

1.9 Interfacing Clearstyle with Clearview

This shall be via a generic file format, the file format must use an Open Standard widely accepted format such as XML. The chosen format must provide data and image design separation, provide the extensibility needed to provide additional real-time and specialist functions.

It is proposed that XML combined with XSL (Style Sheets) could be used as the basis. A DTD (Document Type Definition) would be generated to provide the specialist functions required by real-time in-plant coders. Being web-based this technology provides methods for linking into Active Components to further enhance the overall solution. See the additional notes provided on XML in section 1.12.

1.10 Building Databases with ClearStyle

Some users will not have suitable existing sources of data for inclusion on the printed images. The Claricom offering must include

- automatic generation of database structures and data entry forms based on Image designs;
- provide flexibility in the way information is linked and how data is located at run-time;
- provide options for:
  - normal linear search of product code;
  - next style selection, where a supervisor pre-selects the products to run on a given line and provides preferred tracking information;
  - customer specific tree style selection, e.g. Red+France+Model 27.

Please refer to section 1.8.2.2 - Automatic database functions for further details of the required database functionality.

1.10.1 The Database Linking Structure

It is important that Clearstyle allows data to be shared easily between the different image designs that will be needed for the various stages of the packaging process. To assist in this it is preferred that the Claricom managed databases are not dedicated to individual image designs. The following process for attaching objects on an image design to fields within the Claricom managed database will function as follows:

1. an image design is created;
2. when an object is selected that requires a Claricom Managed database the user must select, or create, a database for the use of the image design;
3. when a Claricom database object is created either: create a new field within the selected database to contain the data for the object or, select an existing field within the selected database for the use by the new object.
4. the database will require a "registry" to keep track of the image designs that use the database and ensure that the use of each field is consistent, e.g. the type of data held in the field is relevant to the object that wants to use it.

1.10.2 Managing A Product Index

Given the varying product selection options that end-users demand (as listed in section 1.10) there is a need to provide an overall product index that allows operators to select a product code without needing to know what image design is used for each specific design.

This concept needs to allow for a number of coders to used on any one production line and for differing coders to be used from line to line and provides a single access point for remote control by modification of a single product->image reference list.

In developing the required database structure it is preferred that the structure allows end-user data sources to add new products without the necessity to modify additional underlying index table structures.

To define the required database structure it is first necessary to formally define a number of terms:

Line	a Physical production line or any other logical group of coders. A product selection will act on a *line* rather that act on individual coders;
Coder	an individual coding device such as a Continuous Inkjet printer or a thermal print and apply machine;
Coder Class	a group of coders that have similar functionality such that it is practical to use the same *image* definition across the group;
Image or Image File	the definition of the style of what is printed on a given *coder* or *class of coders*, e.g. the number, position and type of objects printed on the image;
Product Code	an unique identifier used to set-up a *line* the product code is used to locate the required data for each printed image and to define the *image file(s)* used for a particular *coder* or *coder class*;
Parameters or Parameter	the definition of what coder performance and configuration parameters are required.

File	
Image Objects	The individual elements that combine to define a particular *image*, these would include text, barcode or date and time objects. Each object will define the source of the data for the object; this data could be generated automatically, such as for dates and times or provided by external data sources;
External Data	Pre-existing databases or other sources of data, for example an end-users SQL Server database or Excel Spreadsheet.

End users do not what to worry about what image file is used for a particular product code at run-time, they simply want to make a single selection for the line.

To provide this across a rather of differing coders and line configurations it must be possible to build or specify a single repository for the product codes that can be run on a given line. This repository or [Master Product Index/Common Product Index/Product Index File] could be specific to individual lines or shared across all the lines in a given factory. In addition the index can be constructed using existing External Data sources or may use a database created and managed by Clearview/Clearstyle.

[Line Configuration File]

This element is really part of Clearview and not relevant to Clearstyle, however, it forms an important link in the definition of the overall database approach, so is included here for completeness.

The file defines the Coders allocated to the line, for each coder:

- physical addresses of each coder;
- the type of coder (Coder or Coder Class);
- Location of Product Code information within the [Master Product Index,...] for this coder, or acceptance of a overall default location;
- Location of the Image File definition within the [Master Product Index,...] for this coder, or acceptance of an overall default location, or the specification of a single default image file for **all** products referenced;
- Location of the Parameter File definition within the[Master Product Index,...], or the acceptance of an overall default location, or the specification of a single default parameter file for **all** products referenced.
- The default path for the image and parameter files for this coder, where the [Master Product Index,...] does not define the complete path itself.

[Master Product Index/Common Product Index/Product Index File]

This is the Claricom Managed version OR a "Deferred" External Database and defines:

- Product Code;
- Image Name (A simple ASCII[6] name or full file name);
- Parameter File Name.

The Line Configuration may negate the need for Image and/or Parameter files by specifying that a single common image/parameter file is used for all products on a particular line.

[Image File]

[6] ASCII is used for simplicity support for other character definitions such as UNICODE etc. must be catered for.

The image file itself will define the links to specific data based on the Product Code obtained from the [Master Product Index...] or by keys derived from this Product Code within the image file itself.

[Form Files]

These objects consist of HTML pages associated with a particular database and used to provide the user interface for collecting operator entered information. This flexibility allows users to tailor data input screens to their own exact requirements.

1.10.3 Differing Coder Capability - Classes of Coder

Any given type of image can be printed by a range of technologies within a factory, however it is usually the case that the style of image is dictated by the lowest-common-denominator coding product.

By allowing images to be defined for a class of coders the benefit of having a generic solution can be maximised.

The following list is provided as an example of how the classes of coders could be constructed:

- primary coders:
  - small character inkjet;
  - thermal over-printers;
  - lasers.
- secondary coders (case coders):
  - thermal label printers;
  - thermal label printer-applicators;
  - hi-res large character ink-jet;
  - large area thermal over-printers.
- tertiary coders (pallet labellers):
  - thermal label printers;
  - thermal label printer-applicators;
  - office printers using specialist label stock

This list does not include large-character inkjet which tends to be used in a hybrid way as a secondary coder but with low-end primary-coder capability, e.g. text only no bar-codes.

The capabilities of any specific coder or class of coders will be provided by the coder-driver layer of Clearstyle; this functionality will be shared with the protocol translation functions of Clearview.

1.10.4 Printer Specific Fonts

Clearstyle must deal with the following fonts:

- True type fonts; used to provide a "best-fit" approximations to built-in coder fonts when generic image designs are being used;
- True type fonts; as "best-fit" approximations to built-in coder scaleable fonts;
- Windows bit-map fonts to represent built-in coder matrix fonts;
- specific scaleable font technologies such as Type-1, Bitstream or equivalent. In cases where coders utilise licensed font technologies the coder manufacture will provide access to the required technologies/licenses etc or Type-Type best-fit equivalent fonts will be used.

It should be noted that for certain classes of coders exact font matching is not essential, slight differences in small-character ink-jet fonts should not cause end-users operational difficulties. Wherever possible data concerning the fonts and character sets used will be sought direct from the coder manufactures.

Issues that are preferably, however be addressed are:

- font size, it is preferred that the required information, as generated at product selection time, can be printed within the print area of the selected coder;
- character sets, it is preferred that characters, in particular European characters (é, Ç Ð, ø, å etc) are mapped from the Clearstyle character-set into the correct code for the target coder. Note, Clearstyle will use Unicode for all internal data storage and Clearstyle and Clearview must handle custom character sets for specific coders.
- adding specific fonts for coders where the fonts can change from one version of the coder firmware to another must be possible using the normal Windows interface. It should not be necessary to edit the Windows registry or other configuration files manually to accomplish this.

It must be possible to enter extended characters easily, the provision of a "Virtual" Keyboard on the screen would allow this.

1.10.5 Bar codes

As with text field handling, there is wide variation between the coders available as to what bar codes are supported. In addition to the firmware capabilities of each coder the differing print resolutions offered by the differing technologies can limit the bar codes that can be printed by a given coder.

Where limitations are due to simple firmware issues then, where possible, Clearview will allow supported bar codes to be rendered as bitmaps. If doing so will limit printer functionality, or throughput then the user should be made aware of this through the general Image information made available when the file is saved.

1.10.6 Printer Resolutions

Printer resolutions vary widely across the range of coding equipment available, from 300dpi, or greater, thermal printers to low resolution valve-jet printers.

Clearview needs to handle these variations in coder resolution. This will present most problems when attempting to use a generic design on a low resolution printer. In such cases it may not be possible to render very small fonts, or detailed graphics.

Other Issues:

- resolution issues must be handled by the coder or coder class driver;
- when designing for a specific coder or coder class the WYSIWYG Image should reflect the effects of the coder or base coder class resolution;
- in the case of ink-jet printers it may not be possible to map pixels to linear dimensions due to the affects of the product speed or slanting of the print head. In such cases an approximation should be given and a warning provided,

1.10.7 Real-time capability

Typically Industrial coders will have certain real-time capabilities provided within the coder firmware. Wherever possible the real-time capabilities used within Clearstyle will be mapped into these real-time functions. This will not be possible in all cases for example: a coder may

not support a particular format of the date, e.g. dd-mmm-yyyyy, 07-Sep-1999 or may not allow an incremental value to be merged into a bar code. In such cases the coder driver portion of Clearview will provide the functionality required. Consequently, this will reduce the maximum throughput of the coder.

Wherever a list of real-time functions is provided the Clearstyle will indicate which functions are supported by the coder's firmware and which are provided externally.

Coder Independent File Format (CIFF)

1.11 Objectives

1.11.1 Openness

The CIFF must be a Open standard that is:

- based on a published and reasonably stable base e.g. HTML/XML;
- able to be processed using a thin client, e.g. a better scenario than using a Cray to process a Level 3 Postscript file sufficiently quickly;
- extensible using software components, e.g. data can be provided or modified or additional imaging functionality added, such as new bar codes symbologies, though the use of additional software modules;

These requirements are born out of the following requirements:

- the need (in time) to allow third party software design packages to be used with the Claricom communications and control solution (Clearview);
- the desire to sell the Claricom concept to coder manufacturers and for them to use the Claricom file format as their normal means of defining the printed image (c.f. Adobe Postscript);
- the need for differentiation.

1.11.2 Classes of Coders/Specific Coders

To allow improved handling of generic coder classes, or specific, coders, the CIFF must provide the capability to store information about users choices, made at design time, regarding how to best "map" generic functionality to specific coder, or coder class, functionality.

In addition it must be possible to provide a means of passing raw coder specific data. For example, a specific customer may have particular functionality built in to special firmware. Ultimately this could be handled using an extension module, however to limit the amount of specialist module programming required, particularly early in the product development cycle, this additional functionality must be provided.

1.11.3 Parameter/Image/Data (Job/Format/Data) Issues

The file format needs to deal with three distinct types of information:

- Parameter information such as print darkness, print speed, print head configuration etc.;
- Image information, e.g. number, position and type of objects included on the printed image;
- The actual data printed, in the style set by the Image data, as previously described this data could come from a wide variety of sources.

Wherever possible the same fundamental file structure should be used for all three data types. It may be possible to use hyperlinks (or an equivalent method) to link parameter files to specific image files. In such cases care must be taken to ensure that it is possible to use differing parameter files for differing coders that use common image and data information.

1.11.4 Extensibility

The ability for the CIFF to allow functionality to be added without fundamental changes to the underlying file specification is preferred. For example, the ability to add a specialist date rounding algorithm or a new bar code symbology.

It is predicated that the use of software components provides the best method for giving this extensibility. A component, or separate components, can be provided to give both the design-time information about the component and the required parameters and the run-time output needed to communicate to target coders.

Components could be generic, coder class or coder specific.

The required component interface will be made available to allow coder manufacturers or other third parties to produce components.

1.12 XML as the Chosen Standard

XML is designed to allow the "designer" tags required to deal with the specialist layout requirements needed (bar codes etc). Adding tags to an HTML file, as above, would be open to the risk of the controlling body for HTML using the tags chosen for other purposes.

It is proposed that Clarlcom would develop the Document Type Description (DTD) for coder images. This DTD is the definition of the specialist tags required by the document (code or label, in our case). Even if the detailed limitations of XML prevent its use as a common open standard for the description of coder images then the addition of specialist component extensions to this language would provide the best compromise solution. Note that once defined this format could be provided to a range of Label Design package companies. For example, Teklynx could provide a Clarlcom XML driver.

As with many open standards it may be necessary to provide the ability to store some native information under the umbrella of the "open standard". This loop-hole allows a user to access features that may be added to a specific coding product just for them, or to access a feature that is not part of the generic features of that family of coders. This method may also be significant in allowing coder configuration information to be stored alongside the image design. For example, a specific product uses the TESCO style of image and because the image has to be printed onto special TESCO material the coder needs to have particular settings of certain coder parameters. It is hoped that much of this information can be stored generically, however, coder manufacturers will continue to innovate in this area and the ability to handle new types of parameters must be supported.

1.13 Information to be Stored

Overall image information is preferably stored, for example:

- overall image size;
- allocation of sections of images to print heads - where appropriate;
- the nature of the design e.g. generic, coder class or coder specific;
- links to parameter information;
- links to databases and other sources of data;

The following object types will be supported:

1 text fields including data:
a) specified at design time and therefore, fixed by the image design;
[illegible] built in database;

c) linked to on ODBC, or other external, data source;
d) a combination of data from other objects contained within the image design;
e) data entered at the time the product is selected for printing; this mechanism must allow a single data entry screen to be used for a number of images; (see section: 1.18.3)

Properties:

- object name for cross referencing purposes;
- position (x,y in generic units, e.g. millimetres);
- font as a generic font name, a coder/coder-class specific font alternative to be provided;
- size (independent of coder technology, note that secondary information can be stored to aid in mapping generic size information for specific coders or coder classes;
- orientation (see footnote 2 on page 9);
- expansion capability for colour etc.;
- object logging information, e.g. Is this object to be logged to file?

2. bar code fields; both conventional and 2D bar-codes; design-time and run-time functionality should encapsulated within software components;
3. graphics;
4. specialist text fields, including:
   a) Time and Date in multiple formats; it must be possible to combine each basic date/time element in any format and where formats are not supported by a specific coder this should be indicated;
   b) SSCC;
   c) Increment and decrement objects; combined text and numeric data should be provided with options for setting the starting value (entered, obtained from database) and increment;

Note that specialist functionality should be provided through independent software components that contain design-time a run-time functionality.

1.14 Extension Capability

It is preferable that the file format allows additional design-time and run-time functionality to be added using additional software components.

The design-time component provides connections to allow required object properties to be obtained and stored within the generic CIFF.

The run-time component will use the properties provided to either render the bitmap image required or to provide data for rendering by the basic text or bar code rendering sections of Clearview, or the coder itself.

1.15 Incorporation within Coders

An important aspect of the future strategy of Claricom is to provide tools in the form of software components; SDKs or potentially source code to coder manufactures to allow the Claricom CIFF to be used as the base method of providing image design and data to coders.

For this to be feasible it is important that the CIFF is relatively efficient to process, preferably by the classes of processors used by state of the art coding equipment.

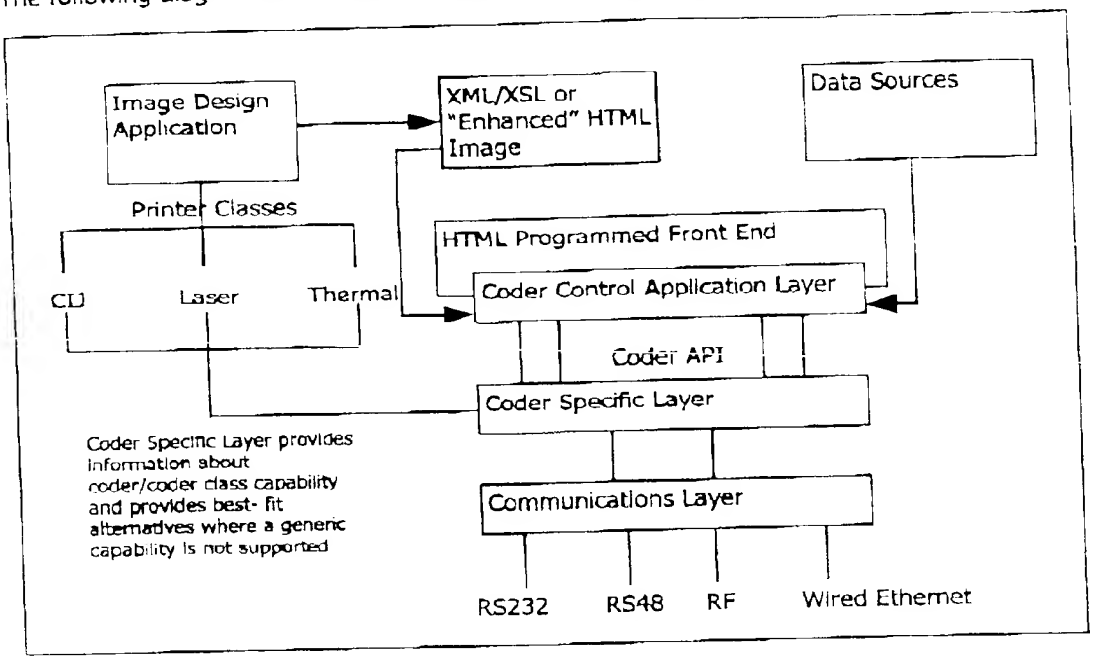
By providing significant amounts of CIFF processing software in the form of separate components it will be possible for specific coder manufactures to control the amount of

software required for their specific application. For example, a 7x5 matrix CIJ or laser coder would not require scaleable font or barcode rendering capability.

ClearView

1.16 Overall Approach

The following diagram shows the basic elements and data interactions of Clearview.



Each layer of functionality must be useable in its own right, for example a SCADA or MES supplier would only use the Coder and Communications layers and would interact with these components using the Coder API.

1.17 Product Layers

1.17.1 Control Layer

This layer provides the interface to the user, top level functionality (product selection etc), provides the configuration elements and allows the extension of "canned" performance using bolt-on software modules.

It is envisaged that the look and functionality of the user interface will be highly configurable though the use of HTML or an equivalent method.

1.17.2 Coder API Layer

This layer is actually the defining specification to the next layer down, the Coder Specific Layer. The Control Layer will use common methods and properties for communicating with the range of supported coders, this coder API will define these methods and properties.

It is this layer that will provide the published common interface for external applications.

1.17.3 Coder Specific Layer

The Coder Specific layer consists of a number of components, one each for the classes of coder and specific coders supported. It is these components that will:

- convert the generic methods and properties into the appropriate commands for the target coder;
- add functionality missing from the target coder;
- providing functionality that is available within the coder but that has limitations compared with the generic standard;
- handles the mapping of fonts and character-set information.

These components (or companion components to the Clearview components) will be responsible for passing design-time information back to Clearstyle.

1.17.4 Communications Layer

The coder layer will only provide the data to be transported to the specified coder. All communications needs will be provided by the communications layer. As with the Coder Specific Layer this layer will consist of a number of components providing a variety of means of physical and logical communications, e.g.:

- TCP/IP over Ethernet;
- Vendor specific RS485;
- Virtual RS232 connections using RF Modems, or equivalent;
- etc. etc.

1.18 Operator Functionality

The user interface "look and feel" and functionality will be configured using a web-page approach. It is recommended that SDKs are used that leverage existing browser technology wherever possible.

The out-of-box solution must allow typical applications to be addressed without recourse to web-page programming. Standard objects shall be provided to give commonly used functions. The default look-and-feel should match latest Windows or Web fashions; taking a browser approach should simplify this. Options for an both an explorer tree approach *and* a graphical representation of a line should be provided.

The sections that follow list the functionality Clearview will ultimately provide as an out-of-box product. Note that any interim design choices made to serve early customer orders must not preclude the later incorporation of the functionality listed.

1.18.1 Configuration and Set-up

1. Provide set-up "wizards" to allow the type, number and configuration of the connected coders to be determined. Wherever possible this process should be automated, e.g. Clearview would be told the types of network connected and would scan the networks to determine what is connected.
2. The set-up process should simply the selection of standard components for use with a particular coder or line. For example a simple check list should allow to mimic display to include a beacon, information about the product running, line speed etc.
3. Coders can be grouped into lines; with line groups the entire group can then be set-up with a single operation.

4. The user interface can be configured to suit the needs of a particular end-user; this includes allowing the product selection to be altered.

1.18.2 Status Monitoring

The following status components shall be provided:

1. a beacon mimic this will provide a graphical representation of a three colour beacon showing fault, warning and OK status;
2. current product information, the current product running at the coder and the number of prints made;
3. product speed information, averaged to give "current" speed and a longer term average;
4. efficiency, giving percentage running time averaged over current hour, current shift, current day etc.;
5. consumables monitoring, e.g. ink, solvent levels, ribbon usage etc. where direct feedback is not available from a specific coder estimation techniques are to be used;
6. for coders with reject capability the number of rejected codes.

All status monitoring components are to provide exception based reporting for unexpected conditions, for example if the speed drops below a pre-set figure or the percentage of rejects is above a given limit.

1.18.3 Product Selection and Data Collection

The selection of a new product for a particular line is *the* main day-to-day user requirement of Clearview. It is this function that will be used most frequently and must be easy to use and intuitive.

Key elements of this are:

1. Product Selection and data entry must be for a LINE. All handling of different data and image design requirements for differing coders on the line must be performed by Clearview.
2. Where a single PC is being using to control a number of lines, either with or without Setpoint boxes being used for the actual control of individual lines, it shall be possible to view a summary status for the whole line and drill-down to get more detailed status of the elements that comprise the line.
3. Three basic forms of product selection are require:
   a) Select for a single product list;
   b) Select for a pre-entered list of products for the day/shift etc (NEXT);
   c) Tree style selection e.g. Type + country + model;
4. The ability to modify or extend the user interface provided by making the interface between the production selection components and the main status polling and reporting sections of Clearview distinct and open.
5. It must be possible to provide automated access to the product select and data entry functions. This must allow external MES programs to control what is printed by a line or to allow extensions such as automatic selection by PLC, selection via bar code scanner etc.
6. The time taken to select products must be minimised and all steps possible taken to reduce the number of disk accesses and the complexity and number of SQL statements used to access information.

Let's look at each of the "standard" production selection methods individually, this will introduce the concept of a "[Data Entry Form]" that combines the manual data entry [illegible] particular line:

Linear Product Selection

Here a single product index provides a list of all the products that are available to run a specific line. The Image(s) uses are stored within the index used but need not concern the user selecting the product. An option should, however, be given to allow product codes to be grouped by Image design.

The list of product codes will be presented alphabetically or in insertion order (e.g. in order of record number) this allows users to group similar codes should they choose[7].

It must be possible to search the list of product coders using the arrow keys and scroll bars or by typing information directly. The ideal solution would subset the displayed list of product codes as more information is typed. E.g. entering A34 would list all product coders commencing with A34... Where only a single product code remains it should possible to select the product by pressing ENTER or the RETURN key; it should not be necessary to click an OK button or physically select the single remaining product code in the product list using the mouse.

Having selected the require product code Clearview will determine what format is required by each coder. The [Line Configuration File] will define where the Image file for each coder, the required record is located using the product code as a key.

Having located all the required format files it will be possible to locate all data required. At image design time it is intended that the user will have specified a single "database" as the source for all operator entered information for all the coders used on the line.

The single database will provide a single "HTML" page to allow the required user-entered data to be set. The base HTML page will be generated automatically by Clearstyle, but by using a programmable approach, such as HTML, it is possible to modify the way the page is presented or restrict the values that can be entered by range or by using a list.

The overall component based approach used with Clearstyle and Clearview will also allow other components to provide data at the product selection point. Clearview will request data from such components, the software within the component will determine if current data is provided or an external operation is performed to obtain fresh data.

Once all the required data has been obtained individual product selections and data transfers to the constituent coders on the line will be performed. The user will be informed as to the progress of the process and the ultimate success or failure of selection process.

NEXT Style Product Selection

In certain applications the products that are to run on a specific line, within a specific time frame is know ahead of time. It is rare that production is so well under control that this can be used without the option to allow a given customer order to jump the queue or to insert a new order, at the last minute. However, this type of product selection is commonplace and by providing an "out-of-box" solution many more end-users could adopt this methodology in order to reduce data errors.

Next style selection is similar to the normal full product list approach, but has two distinct stages:

1. Supervisor or automated pre-selection.

[7] Management of this sort of user controlled record order within the context of any relational database is highly difficult but this requirement should be considered wherever practical.

The list of products that is valid for the selected timescale is provided, either by manual selection using the linear product selection method outlined above or by an external application providing a index and data.

Note that in both cases either all "operator entered" data will be pre-set, or some information will be pre-set with the exception of information really not known until the time the product is selected, such as batch code. No products are actually selected at the line by this process, the product codes and data are simply banked for use in stage 2.

2. Run-Time Product Selection

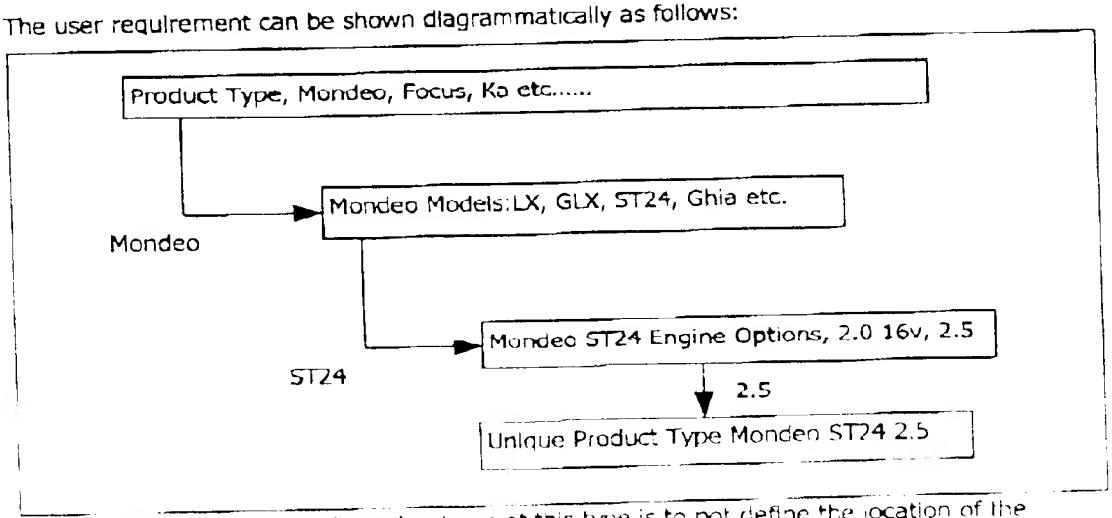
When a new product is to be selected for a given line a line operator will use Clearview to select one of the products selected during stage 1. If all other information has been entered no further operator intervention will be required. If data for a operator entered field has been omitted then the line-operator will be prompted for such data in the same way as for the linear product selection method. To simplify this process a simple NEXT button should be provided to allow the next item in the list to be selected without selecting from the list directly.

Once the required product has been selected the remainder of the product selection process continues in the same way as for the linear product selection.

Tree Style Product Selection

This method is more complex as it requires a hierarchical means of grouping product coders and does not, therefore, fit-in with the simple master product index methods that can be used for the first top product selection options.

The user requirement can be shown diagrammatically as follows:



One method for navigating tree structures of this type is to not define the location of the product index but to allow this to be selected by the user at product selection time. This allows a directory tree to be used to provide the hierarchy. E.g. the root directory has one sub directory for each product type, the Mondeo subdirectory has an index file for each model that contains the product coders for each engine option on that model.

It is assumed that it will be acceptable to provide a unique product ID for each combination to then be used for data location etc for the selected product. Once this unique ID has been [illegible] product selection would proceed as for the other two methods.

Coder Driven Product Selection

In addition to the top-down product selections outlined above Clearview will also provide the ability for a given coder on a line to act as the product selection MASTER. This method can only be supported when the specific coder's protocol provides facilities to allow this.

1.18.4 Data Logging

Providing connectivity between coding equipment and other IT infrastructure makes it possible to use the status and counts information from the coders to provide production logging information.

Clearview needs to provide data logging of all available data particularly with dynamical updated information (where this is technically possible). For example, it will be possible to log increment values and automatically generated sell-by-dates; where these operations are performed by the coder this may only be possible if the coder's native communications protocol provides access to the information. However, the overall application structure must allow for this class of information to be handled.

It will be possible to select what information is logged within an image design and provide additional information from Clearview, e.g. status, product counts, line speeds, time of event etc.

At a detail level all inter-component messages for logging purposes should be sent on an exception basis. E.g. the information being measured has changed.

It will be possible to access logged information at two levels: at the Coder API layer where raw data will be presented or at a file level from the Control Layer. In both cases it will be possible to select when data is logged on status change, on a timed basis. In the case of the Coder API interface it should be possible to select a polled mode of operation.

The file based interface from the Clearview Control Layer should include ASCII CSV style and should reference any applicable formats of CSV file that are considered as "standard" within the SCADA community. It shall also be possible to log data directly to a ODBC or DAO data source.

1.18.5 Handling Errors and other Exceptions

Wherever possible Clearview should report errors without stopping any additional polling or reporting functionality. Only in extreme cases of unrecoverable program errors should the application stop.

It should be possible for Clearview to be configured to allow errors to be posted to a log file and ignored or for certain errors to halt operation of a given line. For example, in pharmaceutical applications data errors would need to prevent the operation of the line. E.g. Stop an report, warn and report, best auto-fix and report.

1.19 User Interface Programming

Customers will often demand that the look and feel and detailed functionality of the user interface are tailored to their specific needs.

To allow this in an extensible fashion it is proposed that the user interface is provided as a series of linked HTML pages. This allows the look and feel of the user interface to be highly

configurable. In addition, taking this approach allows additional high-performance functionality to be provided using Active components and gives the maximum flexibility to connect to wider networked systems and view the display from other points on the network.

The out-of-box product will provide commonly required functionality as a series of canned pages. It will be possible to configure and use these canned pages without any knowledge of HTML. It must, however, be possible for suitably trained personnel to access the base HTML pages and change then, make additions or completely restructure the interface.

1.20 Built-In Database Functionality

As part of the out-of-box functionality it is preferred that Clearview provides simple methods for accessing the Clearstyle database. This will be in the form of pre-coded components that provide the product selection methods described earlier in the section.

It is important that the performance of these built-in functions is as swift as possible. There is a clear linear relationship between the number of applications that can be undertaken using "standard" software and the speed of the built-in database searches.

1.21 Client/Server Operation

Given current trends within IT, as a whole, and specifically within industrial control, for distributed solutions, it is important that Clearview provides Client/Server style operation.

The functional requirements are:

- it shall be possible to run Clearview on small industrial workstations that provide local connection to the target coders, a shop floor user interface and upward connection to the wider IT network;
- it shall be possible to monitor a number of such small Clearview platforms from any point on the IT network and view a summary of the status or "drill-down" to get more detailed status of the performance of a given Clearview application;
- it shall be possible to control any given line from any authorised workstation on the network.

Conversely, Clearview applications running on industrial workstations must be able to obtain data from databases resident on the wider IT network. In some cases end-users may chose to keep local working copies of databases on the industrial workstation to protect against network faults but Clearview must not demand this method of operation.

[Given the nature of the relationship between the industrial PC based versions of Clearview and the IT network resident versions it may be that the terms Client and Server are not useful. The smaller (in PC terms) units based on, or near, the factory floor are most usefully called Clients but in the context of the application act more like Servers.]

1.22 Upwards Connectivity

As described above it is preferred that Clearview can obtain data from data sources based on the network. To allow this the following data access methods will be provided, as a minimum:

1.22.1 ODBC

The CIFF file format generated using Clearstyle will provide the ODBC data sources used and the specific key field, source of the key from within the active image design and the target data field.

Clearview will handle character set mapping from the native database character set to that of the coder. Where no equivalent character is provided a warning shall be provided.

Where a data source is not available or a record cannot be located error message shall be provided that give as much information as possible about the nature of the fault, e.g. name of data source, if the source exists what is the file name and location specified, if the record was not located what field was being used as the key and what was the search value being used.

1.22.2 SQL

This can be provided as a particular application of ODBC, alternatively the MFC may provide direct support for SQL with improved performance.

In addition it shall be possible to use advanced data access methods to specific complex SQL statements to locate the required data. It shall be possible to use of objects from within a particular image design to provide key information.

1.22.3 DAO/ADO

Within the Microsoft programming environment ADO now provides better database access performance than DAO or ODBC. Wherever possible Clearview should use ADO to get the best possible database access performance.

1.22.4 COM/DCOM

By using a HTML/XML as the framework for the Clearview user interface and the CIFF should make the addition of functionality used COM/DCOM modules as simple as it can be.

1.23 Extensibility

It is impossible to foresee all the potential end-user applications of coding equipment and even if this is possible, the detail variations of particular installations will still require tailoring on a case-by-case basis.

To allow this Clearview must allow for additional software functionality to be added without changes to the base product.

1.23.1 Additional Functions via COM/DCOM objects

Additional functionality will be added using Microsoft COM/DCOM model. The interfaces between the components and the main Clearview application will be defined in the HTML that defines the user interface.

It must be noted that in some cases the additional components may be required to perform high performance communications functions (such as interfacing to PLC networks) or similar polling style functions.

Image processing components will also be required these will be referenced as part of the CIFF describing image formats. These components will require two distinct sets of

behaviours: to allow *ClearStyle* to configure to operation of the object using the component at design-time and to provide the run time data or bitmap generation at run time.

1.23.2 The use of COM objects in a Client/Server Environment

To make the use of Image Processing components as efficient as possible the separation of the design time and run-time sections should be considered. That is, can the Clearview requirements of such components be provided independently of the design time (Clearstyle) requirements.

1.23.3 Adding to Coder Functionality using COM objects

An additional advantage of having image processor functionality programmed as components means that coders that implement the Claricom CIFF file format can add new coder functionality as downloadable components. For example, the processing for a new barcode symbology or a bespoke shift code formatting scheme could be provided as a downloadable component.

Allowing design-time properties to be revealed by the Component in Clearview will mean that the fact the functionality is added in this way will be transparent to the end-user.

As stated previously, splitting the design-time and run-time functionality of components must be considered to reduce the amount of software that is needed in a Setpoint box or, ultimately, that is downloaded into a coder.

1.24 Specifying Back-up Coders

In certain types of coding and labelling applications end-users require the system provided to automatically handle the use of backup coders. This is most common in pallet labelling applications where a desktop thermal label printer is used to print labels if the automatic labeller is unavailable, for any reason, or for labelling part pallets.

Clearview will include standard capabilities to provide this, it shall:

- allow an alternative coder to be associated with the primary coder;
- allow the end-user to define the circumstances when the alternative coder should be used;
- handling or part pallet functionality will vary between applications hooks shall be provided to allow a simple component to be added to identify an handle any data entry for part pallets.

1.25 Controlling User Access

As with any program of this type it is preferred that Clearview handles differing types of user and limits access to configuration and programming features to authorised personnel. In addition product select may also be restricted to particular users.

Access control should:

- user Windows user-names wherever possible;
- allow access to be granted to functions on a user by user basis;
- unknown users should be simply allowed to view status;

Clearview will provide the capability to log significant events, it should be possible to configure the level of event logging required, typical events will be:

- changes to the configuration, date, time, user name summary of changes;
- product selections, date, time, product selected, variable data entered;
- program start and stop dates and times;
- unexpected conditions and program exceptions;

Care should be taken to avoid any misleading overlap between these SYSTEM logging functions and the User Level logging functions described in section 1.18.4.

Setpoint

1.26 An industrial PC

- low cost;
- suitable for use on the factory floor;
- support for TCP/IP Ethernet networking to the wider IT infrastructure;
- support for a variety of coder networking options, point to point RF, RS232, RS485, Ethenet etc.;

1.27 An access point for line set-up

End-users of Claricom solutions will come to regard the Setpoint unit as the focus for user interaction with the system *and* as the junction box for connecting the factory floor coders to the wider IT infrastructure.

1.28 A Node for Connectivity

A major benefit of this solution is that is provided a suitable user interface for product selection and status monitoring, where it is needed, on the factory floor. It also allows all the potentially intense communications requirements of the factory floor coding equipment to be isolated from the wider IT network.

Setpoint can, therefore, be more responsive and prevent the wider factory network from being swamped with coder polling traffic.

In addition, by allowing the "PC" to be located on the factory floor wiring costs, or RF range, are minimised.

Datapoint

Not all coders have built-in protocols and hardware capability to allow networking. In cases where a stralght RS232 connection is not practical additional hardware and software is required to provide a solution.

Datapoint will provide this. The Datapoint boxes will be small and cheap and provide solutions for:

- point to point RF modems;
- Claricom RS485 (see the following section on networking);
- TCP/IP Ethernet;

In at least two of these applications externally sourced and badged hardware will be used.

The Network

Not all coders will have networking capability and in some cases end-users will demand a single networking solution rather than simply bridging between differing coder vendor networks.

To meet this need Claricom will need to provide a range of networking solutions. These will include:

- Ethernet (TCP/IP using Sockets) using wired or wireless physical transmission;
- simple point to point RF modems;
- a Claricom RS485 solution;
- the ability to use certain Coder Manufacture protocols;
- the ability to use PLC communications methods (Bitbus, Profibus etc).

It is important to note that the limitations of PLC networks will restrict functionality. However, many end-users only required limited functionality and have already invested heavily in PLC networks and are very reluctant to install additional networks. By allowing the communications layer to support variable packet sizes and by providing feedback when a request results in bursting the packet size, Clearview can provide the best possible support for these types of network.

1.29 Dial-up Support from Day one.

Support preferably provides:

- modem access via a desktop PC on connected to any Setpoint units.
- PC Anywhere style access to the PC to allow operation to be monitored and configuration information to be accesses.

11. Software Specification

In general, the case coder software preferably uses the latest modular programming methods in order to maximise the re-usability of software modules, minimise the complexity of code maintenance and enable the most cost effective methods of code customisation and updating for customers. Similar principals are found, by way of example, in the Microsoft COM and DCOM standards.

The control software and the user display will create the lasting differentiation in this and other future products. It is therefore intended that the architecture will allow the incorporation of Web based elements to facilitate :

- On-line configuration and system maintenance via the Internet, Company Intranet or direct modem connection
- Implementation of easily configurable user displays via downloadable Web browser pages
- Compatibility with the system architecture being created at Claricom
- The use of 'Active' components which can be downloaded from other computer systems in order to create new system functionality

The 'Active' component concept is similar to that used within Microsoft Windows in the form of 'Active X' components and other similar modules. These would be fully functional modules,

inside the coder would act as the 'Container' into which the downloadable 'Active' components are inserted.

The point behind this concept is to allow easy, cost effective and powerful upgrade to the functionality of the coder in the form of downloadable 'plug-in' modules. For example, the development of the software for any coding device is dogged throughout its life by the constant requirement to add functionality, which is often driven by specific customer requirements. Custom batch, date encryption or shift codes are examples of such. New barcode symbologies (eg PDF417) are another. Implementing these changes in the traditional development environment can be expensive, time consuming and incur unnecessary engineering change control and testing. In addition adding such features can create a significant field service and support burden.

The objective of this Active component concept is to eliminate much of this wasted time and money. Any new Active components for executing such features as new barcode symbols or customised field design could be posted on the Intranet or the Web Site and be available for download into target coders in the field without the need for the issue of completely new software versions or visits by service engineers. The possibility of the new feature corrupting existing system functionality should also be much reduced by such a feature.

This concept is particularly beneficial in the field of package coding and labelling devices. It is anticipated that this technology will be incorporated into all the Company's future products and therefore the product design should seek to extend this principal to as many aspects of the machine's functionality as possible and not simply limited to the examples of batch, date, shift and barcode generation.

The case coder should be able to perform all the printing functions normally associated with a thermal printer.

In addition to this, the case coder should incorporate as many remote diagnostics and software update functions as possible. The following features are required as a minimum.

a) System software should be upgraded without the use of removable EPROMS or specialist hardware and software. The update procedure should be possible via a modem and serial link with a direct point to point connection or via the Internet
b) The system should support EAN 8, EAN 13, ITF14, Code 39, EAN 128, UPC-A and UPC- E barcodes
c) The system should support printer resident fonts, vector outline fonts and downloadable fonts
d) The system should support the following field types

    Fixed text
    Variable text
    Time and Date fields to a flexible format
    Expiry date fields to a flexible format
    Shift codes and custom codes to a flexible format
    Counts for batch, total (User resetable) and total (non user resetable)
    Logos (.BMP or other industry standard)

    Note : The flexibility with which special format codes can be constructed by the user without specialist programming knowledge is important (eg ref Markem 9064 custom codes)

e) Database capacity for label layouts, logos and fonts should be as large as currently affordable. The board design should allow for expansion to the next size of RAM chips as they become affordable. A ball park target for capacity would be 500 'typical' labels designs and 100 'typical' logos. Products will be stored and referenced by a product reference code. Products will be selected from the system console by entering this code.

f) An easy method of transporting data between units (in a non-networked installations) should be available as an option (eg PCMCIA)

g) A simple method of "mirror printing" should be available where both sides of the case require identical images. This should eliminate the need to design two identical labels or for the operator to load the designs individually for both sides

h) A method of updating dynamic fields via a serial link should be implemented for weigh scale type applications. Since the nature of the scale interface is often unknown, a simple way of configuring the system to recognise and then strip off the preamble and postamble characters in the serial string is required. An installation engineer should then be able to configure the printer to receive, strip and then print the required data on the case without system software changes.

i) The system should have the ability to receive a simple label selection command followed by an 'update variable fields' message from a simple computer or PLC such that a remote system can dynamically change the data and design on a case by case basis if required. The serial interface should be simple and ideally based on the command structure of Zebra's ZPL which is familiar to application programmers.

j) For such live systems, the coder needs a variety of default start up conditions from power up or error

- Default to printing the last label design that was previously printed
- Default to a blank case
- Default to an error state until new product data is sent (ie if a box arrives and no new data has been sent, an error condition occurs)

In this way the user will have the options to control how the system recovers after an error or power up in order to ensure that cases are not coded incorrectly if this is important.

k) The coder must have a flexible method of defining the action required upon either system error or warning conditions. This should include flexible control of the reject and error volt free outputs to determine

- What events cause them to get set
- How they are set (level or pulsed) + pulse duration
- Sense of the signal (high or low)

l) The software should be configurable for screen language. Language sets should initially include

- English
- German
- French
- Spanish
- Italian
- Dutch

With sufficient system expansion abilities to ultimately include

- Swedish
- Norwegian
- Portuguese

Consideration should be given in the design to how oriental character sets would be handled and the system support requirements for this.

For each language, the system should be capable of time and date stamping in the native language. This feature should be activated by way of a setup parameter.

m) A method of enabling remote dial up is required which allows a remote help desk technician to do the following

- Upgrade the system software
- Upload the entire system parameter map
- Download an entire system parameter map
- Selectively modify the system parameter map 'on-line'
- Upload the entire label, logo and font database (or selected parts thereof)
- Download the entire label, logo and font database (or selected parts thereof)
- View the operator screen at any time
- Access any system diagnostics not available via the operator panel

n) The system diagnostics, which are available via various levels of password protection from the system console should allow the user to see the status of every electrical input and output in real-time and provide the opportunity to force an output on or off from the panel. A test image should be available which is designed to give the best possible feedback regarding jet performance and print quality. It should be possible for the system to print its own parameter set up onto a target substrate of suitable size (ref Zebra printers setup printout).

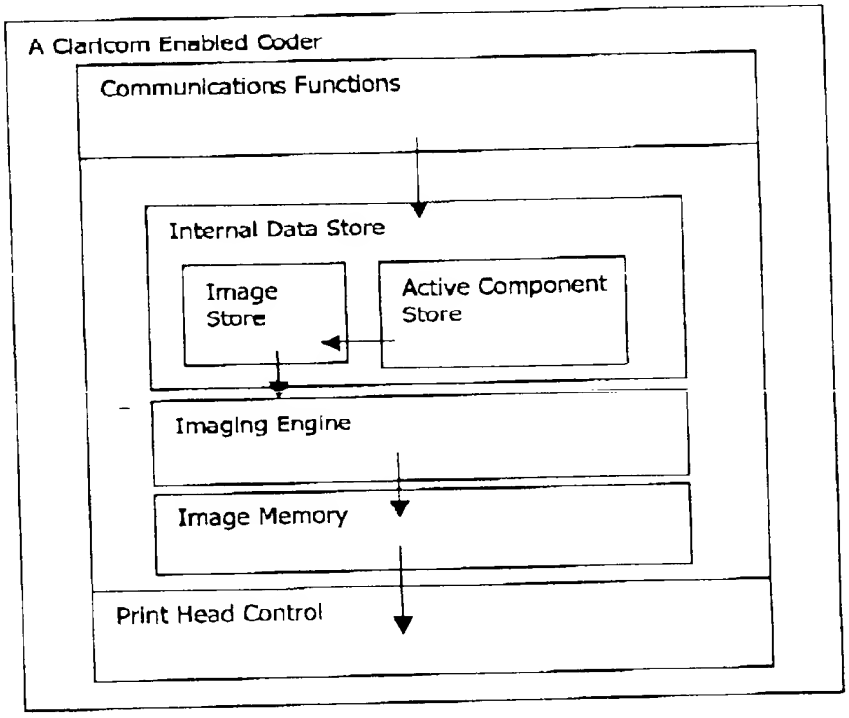
o) The host serial interface to the system should be printable ASCII with minimal checksum calculation that can be disabled if required. A simple ACK/NAK system of acknowledgement and resend request should be used. The interface should provide the following facilities as a minimum

- Read and set the system Real Time Clock
- Select a label design from local memory
- Download the label data for a selected format
- Read the system I/O
- Modify the system I/O
- Serially initiate the printing
- Serially disable the printing
- Read the system counts and current product code
- Reset the system counts (except the non user resetable count)
- Read the system status message being displayed on the console
- Do a product select by sending a line number which is internally cross referenced to the label design

p) A method to print different label images depending on which line number the product comes from. This is activated by either a BCD digital line number signal or via the host serial interface. A method of creating either a lookup table relating line to product, or an array of 'selected' label images, referenced by the line number, is required.

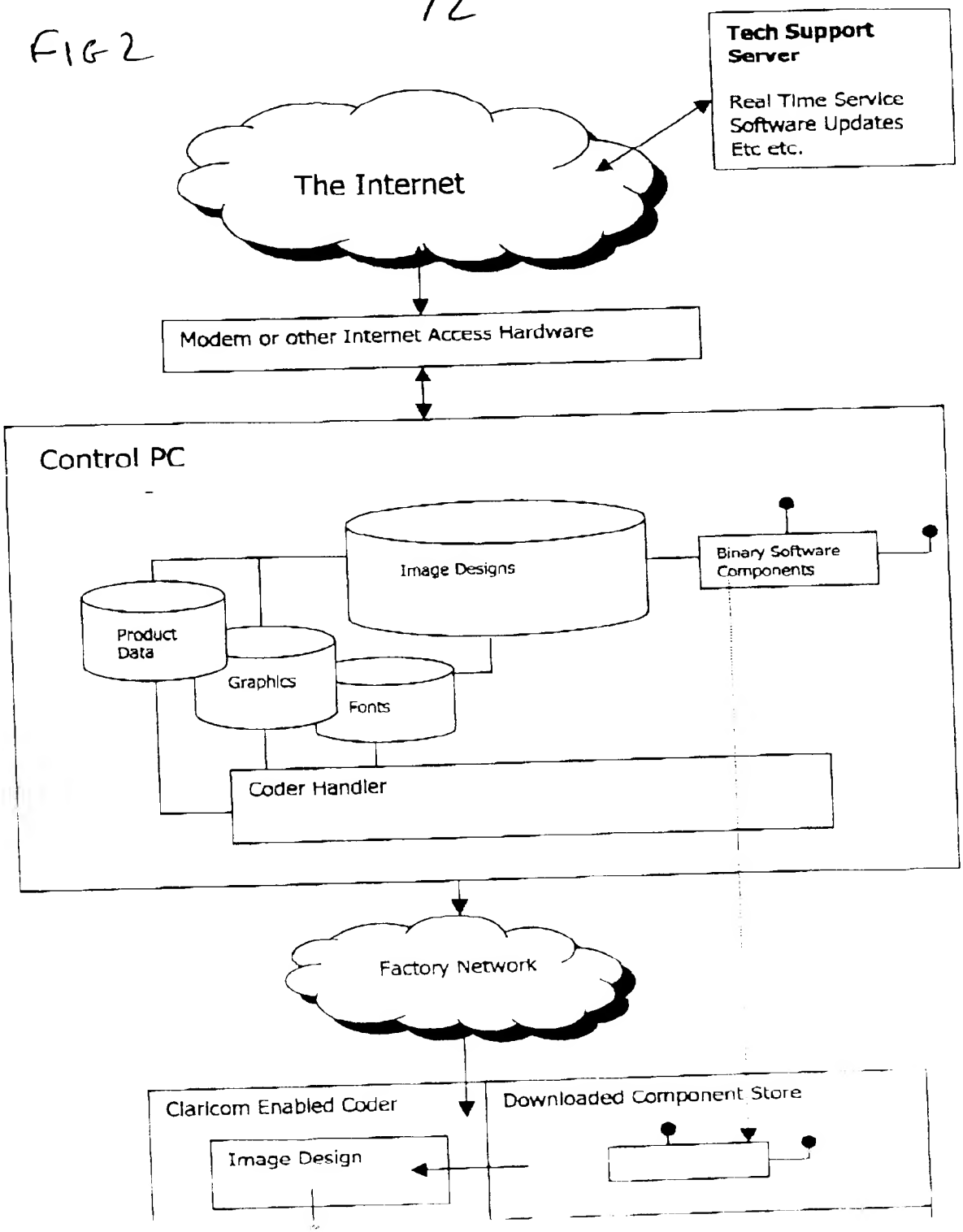
q) The software design package required for creating the print layouts will be either an existing label design package or a product from Claricom. The system architecture must be able to support either. In the event that an existing package is used, the modifications required to drive this new system should be an absolute minimum – ideally none.

FIG 1


A Claricom Enabled Coder
Communications Functions
Internal Data Store
Image Store
Active Component Store
Imaging Engine
Image Memory
Print Head Control

2/2

FIG 2


Tech Support Server
Real Time Service
Software Updates
Etc etc.
The Internet
Modem or other Internet Access Hardware
Control PC
Image Designs
Binary Software Components
Product Data
Graphics
Fonts
Coder Handler
Factory Network
Claricom Enabled Coder
Downloaded Component Store
Image Design